2002

Optimizing Plant-line Schedules and an Application at Hidden Valley Manufacturing Company

Brown, Gerald G.

Optimizing Plant-Line Schedules and an Application at Hidden Valley Manufacturing Company

Gerald G. Brown • Robert F. Dell • Ray L. Davis • Richard H. Duff
Operations Research Department, Naval Postgraduate School, Monterey, California 93943-5000
Operations Research Department, Naval Postgraduate School, Monterey, California 93943-5000
Hidden Valley Manufacturing Company, 1221 Broadway, Oakland, California 94612-1888
INSIGHT, Inc., Sudley North Business Center, 7960 Oonegan Drive, Suite 233, Manassas, Virginia 20109-8236
gbrown@nps.navy.mil • dell@nps.navy.mil • ray.davis@clorox.com • rduff@insight-mss.com

This paper was refereed.

A plant line schedule specifies a plant's sustained batch operations over time with detail sufficient to manage all activities. Plantwide considerations include restrictions on how production centers can be formed from production lines, packaging lines, conveyers, and so forth; the cost and time of product-package item setups, changeovers, and shutdowns; honoring in-stock service levels, minimum inventory, and committed shipments; recognizing efficiency gains with longer batch runs; respecting crew constraints; and the costs of materials, labor, and carrying inventory. We developed a cost-minimizing optimization model, PROFITS, that features multiple independent time streams for various categories of events that mimic existing periodic reviews of operations. PROFITS is embedded in a graphical user interface that eases the grueling aspects of scheduling: preparing data, controlling scenarios, and visualizing results. At Hidden Valley Manufacturing Company, completing an eight-week plant-line schedule takes about an hour. This is much faster than manual scheduling was—and the schedules are better.
(*Industries: agriculture–food. Production: scheduling–planning.*)

We developed an optimization model for plant-line scheduling of sustained, single-stage batch production and packaging of multiple product-package items over a multiweek planning horizon. Most consumer products, with examples ranging from packaged food products to industrial lubricants, are produced and packaged in this way. We seek coordinated patterns of production and packaging that minimize total cost; costs include regular-time production, overtime production, item setup, item-to-item changeovers, item shutdown, and inventory storage, as well as penalties for shortages (unmet demand), safety-stock violations, and ending-inventory violations. The amount of time lost performing item setups, changeovers, and shutdowns depends on the sequence of items produced. For each item, the model tracks beginning inventory, time-varying demand forecasts, committed shipments and times, and goals for safety stock and ending inventory. The plant has limited capacity for storing inventory.

A *production center* is a group of one or more production lines, plus one or more packaging lines, conveyers, and so forth, intended for sustained activity. Firms plan production-center activities in terms of production campaigns. A *production campaign* is a production-center schedule that details, to the nearest minute, events in a sequence of production periods, changeovers from item to item, and idle periods.

0092-2102/02/3203/0001$05.00
1526-551X electronic ISSN

Various production centers, running at potentially different rates, may produce a given item. The production rate of an item may improve with experience over the duration of a batch. A complete *plant schedule* is a recommended portfolio of campaigns, including the exact time of every event.

To select a plant schedule, one restricts the choice of a group of alternate candidate production campaigns and measures their joint consequences by periodically reviewing their collective influence on several types of system states. Reviewing a particular type of state at a particular time gives rise to an *event* of that type. Successive events of a given type demarcate a time *epoch* of that type (Figure 1). The type-one epoch (the most aggregate) is the planning horizon. At this level, we summarize cost, production, and packaging activity, and the satisfaction of horizon-ending inventory goals. At each type-two event, we reconcile production and demand during the preceding type-two epoch with their consequences on the state of safety stock and inventory. At this review, we also assess plant availability at regular and overtime cost, plant inventory storage limits, and for each item, its demand, inventory-holding cost, and safety-stock goal. During each type-three epoch, line usage for both regular and overtime is limited, and simultaneous use of production and packaging-lines is restricted because of limits on crew availability and plant layout. Production centers can be activated or deactivated only at a type-four event. Each type-five event is a planned shipment. More epoch types may be defined as needed.

A production center usually consists of just one production line and one packaging line, but their configurations may be more complex. Engineers develop standards for each configuration's aggregate operation and efficiency: For each production line, they set standard throughput rates for each product and product-to-product changeover costs and times. For each packaging line, they also set throughput rates and costs and times for changing packages, cartons, or labels. For simple production centers, the aggregate rates and costs easily derive from the rates and costs of their components.

For item-to-item changeovers, which include special cases for empty-to-item setup and item-to-empty shutdown, changeover times and costs are sequence-dependent (for example, a changeover from item A to item B may differ in cost and time from the reverse order). Empty-to-item setup times and costs are independent of any prior activity, and item-to-empty shutdown times and costs are independent of any following activity.

In the Appendix, we explain how to generate candidate production campaigns and describe an integer linear program, PROFITS, that selects the best plant schedule from a set of alternate campaigns.

Why a New Model?

Models for optimizing single-stage, multiple-item production and packaging range in scope from long-term master planning to short-term scheduling (Lawrence and Zanakis 1984). Master-planning models use time epochs of weeks or months (for example, Nicholson and Pullen 1971). Linear-programming (LP) master-planning models ignore the fixed costs and delays of setups and changeovers. Integer linear programs (ILPs) for master planning typically represent setup time as a fixed time and setup cost by a fixed charge incurred whenever nonzero production of an item occurs in an epoch (for example, Brown et al. 1981). To limit the number of integer variables, the developers of such models seldom directly capture item-to-item changeovers. These ILP models do a better job than LP models in accounting for multiple-item production, inventory, and limited shared capacity but still do a poor job of representing setups and changeovers.

In contrast, short-term production-scheduling models typically represent every admissible sequence of operations in terms of pair-wise orders and cover a time horizon of just a few hours. Integer linear programs of such representations suffer from the curse of dimensionality: Representing fixed charges and times for item-to-item changeovers requires a binary decision variable for every ordered pair of items. Scheduling models do a good job of accounting for setup and changeover costs and delays. But, because the consequences of independent sequences of operations are awkward to accumulate in a common, fixed-time increment, scheduling models do a poor job of representing limited capacities, especially shared capacities. In practice, to solve such scheduling models, when the

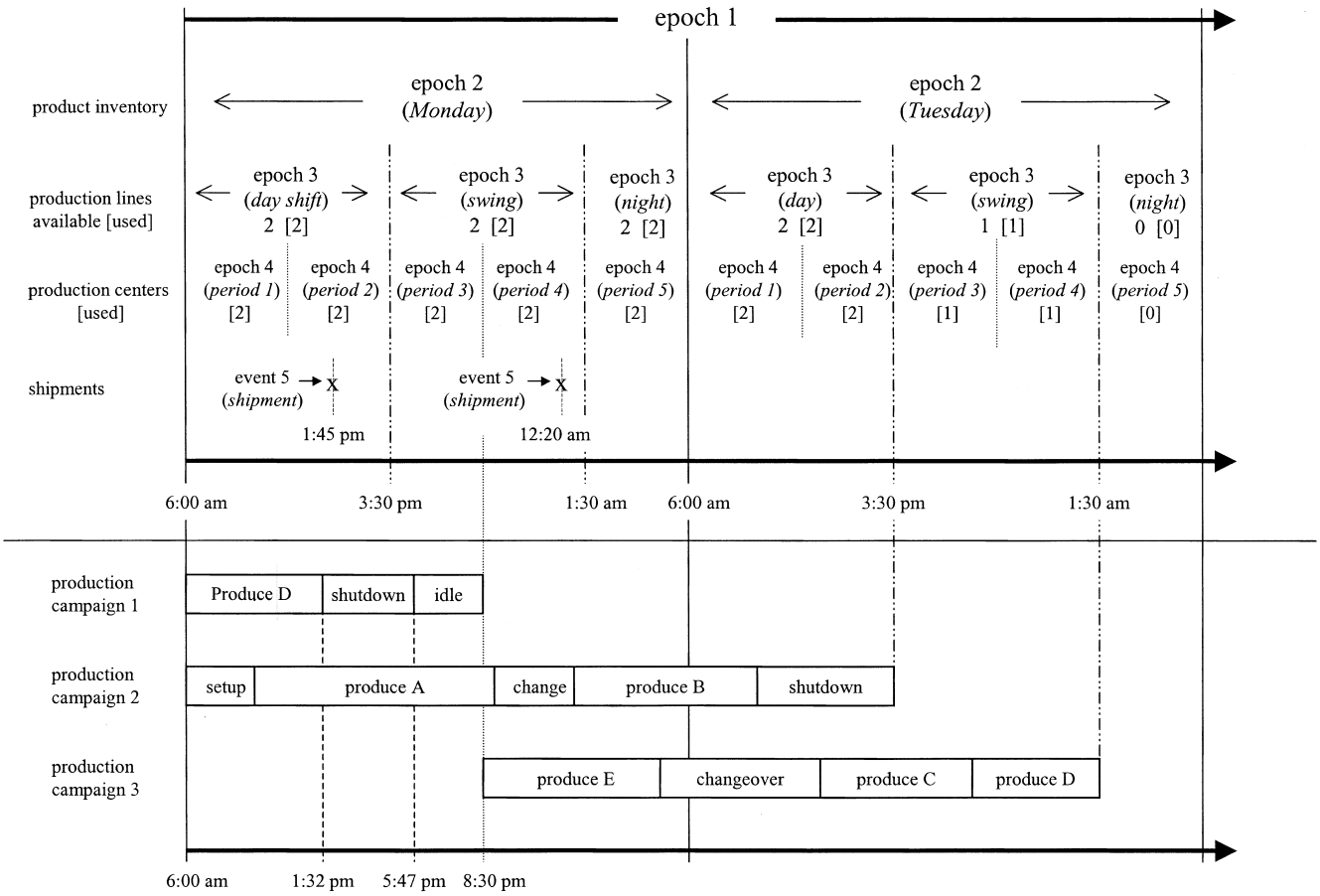


Figure 1: This hypothetical plant schedule consists of three production campaigns. Campaign 1 begins on Monday and starts a production run of two quantity increments of item D at 6:00 am, starts a shutdown at 1:32 pm, becomes idle at 5:47 pm, and ends at 8:30 pm. Five types of events and epochs are shown at the top of the figure. The state of inventory for each item is accounted for only at the transition from one day (type-two epoch) to the next and is based on the inventory at the start of the day, the day's demand, and daily production from the production campaigns. The number of production and packaging lines operating simultaneously is restricted during each shift (type-three epoch). Here, no line is available during the night shift on Tuesday, only one production and packaging line is available for the swing shift on Tuesday, and two packaging and production lines are available for all other shifts. Production campaign 2 starts with a setup, produces item A, continues with a changeover, produces item B, and terminates with a shutdown. Production campaign 3 starts producing item E on Monday at the beginning of period 4 (type-four epoch); production centers can start or stop only at the transition from one type-four epoch to the next. Campaign 3 shows a situation where neither an initial setup nor a final shutdown is required. Two shipments are scheduled at 1:45 pm and at 12:20 am: these are type-five events.

models have more than a few items or have extended time horizons, we resort to heuristics (for example, Kuik and Salomon 1990, Glover and Laguna 1997). Most of these heuristics can't tell you how far from optimal their schedules are, and the heuristics that do offer model-specific bounds typically give only weak bounds. Heuristics are delicate, requiring a lot of hand-crafted tuning, and the best prescription for finding better schedules is of uncertain effectiveness: keep searching.

Master-planning models are inadequate for plant-line scheduling when setups, item-to-item changeovers, and shutdowns are too significant to approximate. LP models cannot represent the essential but concave features of efficiency that improve with batch size. Short-term scheduling models attend to detail finer than needed to manage plant-line operations and do a poor job of identifying and allocating scarce resources.

A heuristic hybrid of LP and a quadratic assignment model (Geoffrion and Graves 1976) has been used to schedule plant lines. This hybrid has been extended to larger scale for other scheduling problems (Sibre 1977).

The bleach-line tool never produced acceptable schedules.

This hybrid is a clever technique, and it can provide good solutions quickly, but it requires a lot of tuning, both of its sequencing heuristic and to control iterations between the LP and sequencing. And it's a heuristic that does not produce a very good objective assessment of solution quality.

The PROFITS model is a multiperiod, multiproduct model for production lot-sizing and scheduling with both local and plantwide resource limitations. The model represents sequence-dependent setup times and costs on parallel, nonidentical production centers that can be configured to operate simultaneously only in limited combinations. The closest model we find in the literature is by Kang et al. (1999), who assume zero setup times, no joint production-center limitations, minimum and maximum lot sizes that are independent of items and production centers, and no plantwide inventory limits.

The variety of time epochs is a distinguishing feature here. Without such a variety, we must massage the data into some universal time increment without overaggregating or assuming greater time fidelity than is justified. This is difficult. Worse, forcing a single time-period duration on a decision-support system inflicts on its users the burden of fabricating data to suit the system. It is much better and easier to provide a timing mechanism for the model to match the time resolution at which each process is managed. In practice, we need only a few types of periodic-review intervals, and although each epoch type can in principle have varying duration and be independent of other types, most epoch types have fixed length, and epoch types align in natural, nested periodic intervals (for instance, hours, half-shifts, shifts, days, weeks).

Real-world production-line schedules are expressed in terms of campaigns, familiar features of ubiquitous Gantt charts (Clark 1922). The scheduler needs direct means to visualize, evaluate, and if necessary manually control most details of a planned campaign. Neither master-planning models, using fixed-length time epochs, nor event-scheduling models, using sequences, express production plans directly in terms of campaigns.

We use time-cumulative goals for production and inventory because a cumulative violation under or over these goals persists over time until it is remedied. Time-cumulative goals promote plant schedules that have "persistence" (Brown et al. 1997): When you can't meet goals immediately, it is desirable to meet them as soon as possible thereafter so that the total effort stays in concert with the overall guidance provided.

We assume that a production campaign will not require too many events. Otherwise, we would have to limit the number of event-permutation schedules that are generated as alternate candidate campaigns, and we might thus overlook a good campaign. Fortunately, this has not proven to be a practical concern. Even a plant operating 24x7 incorporates planned maintenance, safety meetings, weekend shift changes, and so forth into its schedule. In reality, production campaigns are interrupted during planned epochs, and this is a saving grace. Also, eliminating very minute details—ignoring differences that don't make a difference—in admissible production campaigns limits their numbers.

The PROFITS integer linear program is a large set-packing model with many side constraints. (A typical instance has several hundred packing constraints, thousands of side constraints, thousands of binary variables, and many hundreds of thousands of technological coefficients.) The set pack provides the essential combinatoric scheduling advice, while the side constraints govern the usual LP concerns about goals and scarce resources. The power of this model comes

at a price: Instances can be very difficult to solve. Each binary column can have hundreds of nonzero coefficients, and this characteristic is widely believed to be the hallmark of a difficult set pack. Hoffman and Padberg (1993), for example, show that problems get harder to solve as the number of rows and coefficients per column increase, and they say that their hardest problems seem to derive from adding side constraints. We have invested a lot of time learning how to solve such models (Brown and Olson 1994). But it is not lost on us that if we encounter a problem we can't solve, this set pack is ideally suited (with all binary variables) for simple heuristic search: We hope not to have to resort to a heuristic.

Implementation

PROFITS offers some additional features that permit its use in the real world (Figures 2, 3, 4, 5, and 6). Foremost, everything is viewed and manipulated in a graphical user interface and modeling environment built for the purpose. We used the PowerVista developer toolkit (2001) because it offers fast prototyping, good scalability, an embedded database, links to widely available commercial spreadsheet and database packages, user-customizable screens and reports, Web-enabled features, and most important, graphics of excellent quality. Initial development required about a month, and continuing maintenance and support takes a day now and then.

The interface displays demand data, production and changeover data, production center assignments, and complete plant-line schedules in both conventional tabular form and graphically. The interface has the look and feel of a Windows application, but it is designed to support an experienced production scheduler: The displays are dense with function buttons and features reminding the scheduler what procedures must be carried out, offering instant navigation from any point in the database to any other, automatic cataloging of multiple versions of a given problem for parallel analysis and comparison, and function buttons that simplify complicated operations on the data and enable easy manual override of part or all of any schedule.

PROFITS accepts and respects user mandates, such as "produce exactly according to this schedule for the first seven days, produce no more than 100,000 units of this item before the ninth day, and produce at least 50,000 units of this item by the 10th day."

The various displays are all endowed with features one expects nowadays, such as flyover, drilldown, and help wizards. The distinctive tool icons invoke special actions, such as the bulk import of a scenario from an exogenous business system and cataloging alternate plant schedules for the same plant or for many plants.

Preparing a plant-line schedule from the initial import of data to the first complete (candidate) schedule can take as little as 10 minutes when users have already prepared the static engineering changeover data. Initially setting up such engineering standards and refining them takes much longer, but the result is durable.

Hidden Valley Food Products Company

In the 1960s, guests at the Hidden Valley Ranch in the Santa Ynez mountains of Southern California fell in love with its salad dressing. As time went on, many others did too as the ranch satisfied demand for the dressing through a successful mail-order business selling to all 50 states and 32 foreign countries. In 1970, the company moved production of the original Hidden Valley Ranch salad dressing—a dry packet of seasonings to which the consumer adds buttermilk and mayonnaise—to Sparks, Nevada. The Clorox Company bought the business in 1972. Today, it has manufacturing sites in Reno, Nevada, and Wheeling, Illinois that produce a portfolio of brands, including the original Hidden Valley dressing and dips, bottled Hidden Valley salad dressings, KC Masterpiece barbeque sauces and marinades, Kitchen Bouquet browning sauce, and Salad Crispins.

Hidden Valley bottled brands—covering more than 75 items, 30 flavors, and 15 package sizes—are produced in continuous mixing and filling operations. Each process line mixes dry and liquid raw materials into a dressing or sauce ready for packaging. Automated packaging lines fill, package, and palletize the finished products, which are then shipped to retail and institutional customers.

File Edit View Calendars Optimize Reports Windows Help

Demand [DemandList]

Nbr	UPC	Date	Demand Type	Demand Quantity
2742	ITEM 1	11/21/2000	S	1957
2743	ITEM 1	11/22/2000	S	4040
2744	ITEM 1	11/23/2000	S	3640
2745	ITEM 1	11/24/2000	S	10000
2746	ITEM 1	11/25/2000	S	11760
2747	ITEM 1	11/26/2000	S	11760
2748	ITEM 1	11/27/2000	D	800
2749	ITEM 1	11/28/2000	S	15400
2750	ITEM 1	11/29/2000	S	11280
2751	ITEM 1	11/30/2000	D	960
2752	ITEM 1	12/01/2000	S	21160
2753	ITEM 1	12/02/2000	S	24880
2754	ITEM 1	12/03/2000	S	22040

UPC Code	Description
ITEM 1	**light flavor in 38 oz bottle**
ITEM 2	regular flavor in 24 oz bottle
ITEM 3	extra light flavor in 2 oz bottle
ITEM 4	regular flavor in 16 oz bottle
ITEM 5	spicy flavor in 16 oz bottle
ITEM 6	light flavor in 18 oz bottle

November 21, 2000

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	**21**	22	23	24	25
26	27	28	29	30		

Demand T	Description
C	Direct Customer Order
D	Demand from another location
N	Net Forecast Demand
S	**Desired Safety Stock**
T	Transfer Order In
X	Transfer Order Out

Figure 2: This representation of a demand-object configuration shows how input is presented and how one maneuvers through it via button clicks. At the left is a demand list showing, for example, 1,957 units of desired safety stock for item 1 on November 21, 2000. The tool icons at the top left offer, from left to right, to filter out all but rows with desired attributes and respectively to locate on the first row matching a query, sort rows, save the configuration, load another configuration, restore a changed configuration, delete a rule, display a row record for further manipulation, save a change, and close the window. Across the first row, clicking the marked UPC selection offers a directory, shown at the upper right, with links to alternate items. Clicking Date presents a calendar with a selected date, which can be changed within the current month by pushing another date button, or scrolling to prior or following months via the arrow tabs. Clicking Demand Type offers the menu of alternatives at the lower right.

To insure the highest standards in flavor, purity, and sanitation, Hidden Valley spends time and money on certain flavor changeovers.

Because most products have limited shelf lives, the company must synchronize production closely with shipping. Many raw materials have long lead times and short shelf lives: schedules must forecast raw-materials requirements accurately.

File Edit View Calendars Optimize Reports Windows Help

Size Changeover Matrix [SizeCOMatrixML]

CleanUp Time | CleanUp Cost | SetUp Time | SetUp Cost | ChangeOver Time | ChangeOver Cost | Lag Time | SetUp Efficiency

From -> To	ITEM 1	ITEM 2	ITEM 3	ITEM 4	ITEM 5	ITEM 6
ITEM 1		11.30	3.50	3.50	8.50	24.00
ITEM 2	6.70		8.00	4.50	8.50	2.30
ITEM 3	3.50	8.00		3.50	24.00	24.00
ITEM 4	3.50	6.50	3.50		24.00	9.00
ITEM 5	9.50	8.00	24.00	24.00		2.00
ITEM 6	24.00	3.00	24.00	9.00	2.00	

Figure 3: A changeover from item 1 to item 2 requires 11.3 hours, but the reverse requires 6.7 hours. A 24-hour changeover is to be avoided. Tool icons remind the scheduler what procedures must be carried out and offer access to changeover costs and other related data. There is also a package-to-package changeover matrix. Changeovers may also be a function of the production center or common item attributes (for example, package type), and PROFITS provides similar changeover views for these attributes. The total impact of a particular changeover is defined as a function of all such components.

As Hidden Valley has grown, it has invested in process and package technology to gain operational flexibility. For example, it can now produce many product flavors and sizes on various processing and packaging lines, and a single processing line can now simultaneously serve multiple packaging lines.

Production flexibility helps the company to provide good customer service but complicates scheduling. A plant-line scheduler, manually balancing production capacity with raw-material availability, finished goods inventories, and shipping plans, may only have time to develop a schedule that works. By 1996, Hidden Valley recognized that it needed to make scheduling more systematic. It wanted to automate the time-consuming task of evaluating alternatives, enabling the scheduler to function at a higher level in the business and to deliver better and more comprehensive schedules to the production floor.

Clorox has an optimization-based scheduling system for its bleach lines, and it tried to convert this application for use at Hidden Valley. However, scheduling the bottled-dressing lines is considerably more complex than scheduling bleach production. The bleach-line tool never produced acceptable schedules or reasonable run times.

Clorox decided to get an entirely new plant-line scheduling system, customized for Hidden Valley and meeting the following requirements:

—Over an eight-week planning horizon, to schedule production of over 75 different items from more than 30 flavors and 15 package sizes;

—To try to satisfy all inventory-level and in-stock service targets;

—To follow all plant and production-center calendars;

—To consider all possible production centers that combined process and packaging lines, including those with one process and two packaging lines;

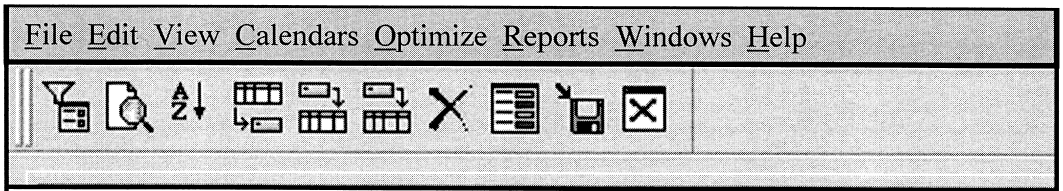
—To respect crew availability;

—To schedule all changeover and sanitation activities in concert with production campaigns;

—To minimize the full schedule cost, including variable production costs of materials, labor, and carrying inventory;

—To make it easy for the scheduler to provide expert advice, express special scheduling requirements, or even freeze all or parts of an existing production schedule; and

—To complete a finished production schedule in an hour or less.



Production Center Results [PCResultsList]

Nbr	Production Center	Item	Date	Period	Amount
7	PC 1	Change 1	11/22/2000	Frozen	4
8	PC 2	ITEM 3	11/22/2000	Frozen	12000
9	PC 2	Change 1	11/22/2000	Frozen	4
10	PC 3	ITEM 1	11/27/2000	1	2220
11	PC 4	ITEM 2	11/27/2000	1	11550
12	PC 5	ITEM 4	11/27/2000	1	6825
13	PC 3	ITEM 1	11/27/2000	2	2220
14	PC 4	Change 2	11/27/2000	2	1
15	PC 4	ITEM 5	11/27/2000	2	9350
16	PC 5	ITEM 4	11/27/2000	2	6825
17	PC 5	Change 3	11/27/2000	2	12
18	PC 5	Change 4	11/27/2000	2	3
19	PC 3	ITEM 1	11/27/2000	3	2220

Figure 4: Production centers PC 1 and PC 2 each follow a predetermined frozen campaign on November 22. In this chronology, production center PC 3 commences a model-selected campaign on November 27 with a production quantity increment for item 1 of 2,220 units. Item change 3 is a cleanup task that requires 12 hours to complete.

Hidden Valley Results

Using PROFITS, a plant scheduler can generate an eight-week plant production schedule in five or 10 minutes on a personal computer. The system considers daily inventory status and honors shift-by-shift crew limits on production-center operations and honors constraints every half shift on utilization of production lines and packaging lines. The optimization usually offers a solution with acceptable service levels (percent of demand satisfied) and inventory levels (percent of desired inventory achieved) that the scheduler can use immediately with only a few minor adjustments. We think the solutions we get are about as good as one can get under current business conditions. A complete planning cycle (from data review through finished

File Edit View Calendars Optimize Reports Windows Help

Item Results [ItemResultsList]

Date	Initial Quantity	Produced Quantity	Demand Quantity	Final Quantity	UnFill Quantity	UnMin Quantity
11/20/2000	1823	0	0	1823	0	0
11/21/2000	1823	0	0	1823	0	134
11/22/2000	1823	0	0	1823	0	2217
11/23/2000	1823	0	0	1823	0	1817
11/24/2000	1823	0	0	1823	0	8177
11/25/2000	1823	0	0	1823	0	9937
11/26/2000	1823	0	0	1823	0	9937
11/27/2000	1823	8880	800	9903	0	5497
11/28/2000	9903	8880	0	18783	0	0
11/29/2000	18783	8880	960	26703	0	0
11/30/2000	26703	4440	0	31143	0	0
12/01/2000	31143	0	0	31143	0	0
12/02/2000	31143	0	0	31143	0	0
12/03/2000	31143	0	0	31143	0	0
12/04/2000	31143	0	2160	28983	0	0
12/05/2000	28983	0	1400	27583	0	0
12/06/2000	27583	0	8520	19063	0	0
12/07/2000	19063	0	3880	15183	0	0

Figure 5: This view of a plant line schedule for item 1 shows, from left to right, date, daily starting inventory, production, demand, ending inventory, unmet demand, and safety-stock shortage. Inventory drops 134 units below safety stock on November 21. A campaign commences producing item 1 on November 27 with four increments of 2,220 units each completed that day. This production run avoids any unmet demand, restores safety stock, and builds inventory sufficient to meet a demand surge that starts on December 4.

schedule) has dropped from more than four hours to less than one.

We ran a blind test between an experienced scheduler (under normal production pressures) and PROFITS (Figure 7). The total plant-line schedule cost was roughly the same for the two schedules. Labor utilization was about the same. However, labor and equipment are used much more efficiently when the schedule minimizes the time lost in setting up, operating, changing over, and shutting down to produce the right

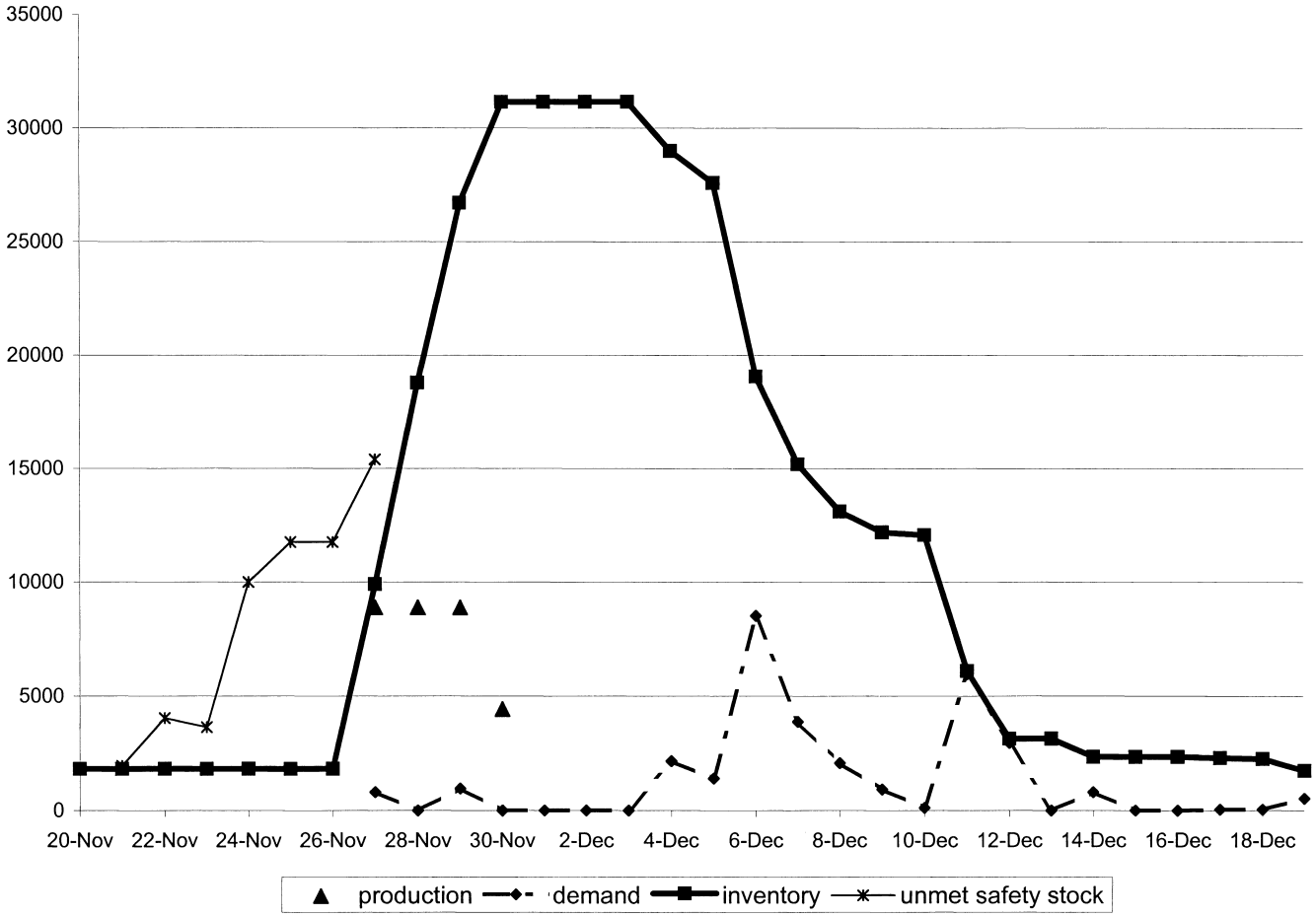


Figure 6: This is the same plant line schedule shown in Figure 5, but this graph makes the upcoming demand surge for item 1 more visible, and it is easier to see the safety-stock shortage that heralds it, the anticipatory production run building inventory, and the inventory draw-down when the demand surge arrives. By the end of this part of the planning horizon, the demand surge has been accommodated and the system has settled back to normal levels of activity. Such ad hoc PROFITS graphical depictions illustrate and thus help explain unusual line-scheduling events.

	Service Level (Percent)				Target Inventory (Percent)			
Item Class	A	B	C	Average	A	B	C	Average
Manual	95.6	87.2	100	96.0	98.8	88.9	248.6	130.0
PROFITS	99.8	97.7	99.2	99.5	106.8	100.6	118.1	108.6

Figure 7: In this comparison of service-level and target-inventory fulfillment for the manual schedule and the one optimized by PROFITS, Hidden Valley has classified its products into A, B, and C items, with A being the most important items. Service level is the percentage of daily demand units satisfied, and target inventory is the actual daily inventory as a percentage of the desired daily target.

products at the right times. Service and inventory levels improve, and the plants react better to the inevitable demand surprises.

Conclusion

Plant-line scheduling is complex, and a good scheduler must master a host of details about every aspect of plant operation. Few schedulers claim to be able to do this daunting task perfectly. PROFITS suggests schedules that aren't perfect, because many details are too mundane to implement in a computer package and are

better left to the experienced scheduler. We implemented the essential details about the costs and efficiency of production via a graphical user interface that makes reviews and changes easy. The underlying model mimics existing review intervals and management cycles and thus reduces abstraction and enhances face validity of solutions. PROFITS invites expert advice and transparently incorporates such guidance. The goal is not to automate scheduling but to schedule better.

Epilogue

The background research, mathematical modeling, graphical user interface, programming, testing, and refining, and everything else we describe constitute about a quarter of the work required to complete an implementation and support its continuing operation. We learned some transcendent lessons from the remaining three-quarters of the work.

When you ask people to do things differently, you need some form of change management. Even if they hate the current arrangement, they will often cling to the status quo as a security blanket. To implement a new arrangement successfully, you must lead through the change and be ready to learn from the experience along with everyone else.

Work to get users comfortable with, rather than intimidated by, computers.

Some decision-support applications are designed for executives, some for managers, and some for operators. Applications for capital budgeting, strategic supply-chain design, master planning and the like are typically designed for managers. In contrast, while plant-line scheduling may have to be sold to executives, it must be used by schedulers, and schedulers have to deal directly with the people actually doing the work. The issues are different, and details really matter. Schedulers know their business, and their success depends on their credibility on the plant floor. They deal day to day with crews and perhaps with union stewards. They cannot, however, aggregate, approximate, or just assume away and ignore the essential details. Employee compensation may or may not be tied to the performance measures you are trying to optimize with the application.

To ensure a successful implementation, you must do the following:

Encourage buy-in from top management—those who will hold you accountable for making the change—as well as from their subordinates. If people don't give the application a fair chance, you may be frustrated and risk failure.

Objectively assess your users and their capabilities. They must be able to follow basic business procedures prior to implementing a new software tool. Document what procedures are in place. If some related procedure is a problem, is missing, or is informal tribal knowledge, you must shore up users' skills prior to implementation. Work to get users comfortable with, rather than intimidated by, computers.

Manage expectations. Be clear about what the application will and will not do. Clarify roles, responsibilities, schedules, and deadlines. Paint a picture literally of what the application cycle looks like—a day in the life. Users included in the development process are much more likely to buy in to the results, particularly if the transition gets tough.

Expect resistance. No matter how open people are to new concepts, when you ask them to depart from established practice, expect some pushback. Patience is a virtue and a prerequisite for success.

Leverage new capabilities. The following situation offers a typical opportunity: A production line is scheduled to run 5,000 cases of item A over two eight-hour shifts, then do a four-hour changeover to run item B. The A run goes better than expected, and the 5,000 cases are finished in only 1.5 shifts. The optimal schedule directs an immediate changeover. The crew argues for taking advantage of the run efficiencies through the end of the shift (an extra 1,600 cases of production). What to do?

If we've done our work right, it should be easier than it was before to visualize and to assess such an opportunity quickly. That so much descriptive information is instantly available with every proposed schedule makes it easy to ponder changes, whether by just seeing the line has capacity or by making another model run. Producing the extra product is not free, and one

can estimate the extra material, processing, and carrying costs. Perhaps not all of the extra product can be sold within its shelf life, and the firm will face return and disposal costs. Why produce a product before it's needed? If the employees are working straight time, the firm pays labor cost whether they produce or not, what's best for the business? The rationale for an informed decision may be instructive for the crew and the scheduler, and it may be used to improve the model.

Anticipate evolution. The schedulers and the decision-support system will improve with experience. Nothing refines engineering standards more than an optimization model exploiting them. Schedulers develop rules of thumb from experience, and they will improve them as they gain insight from the system. Sooner or later, some scheduler will decide that the system isn't doing much better than a manual schedule based on the latest rules of thumb and may be tempted

Schedulers know how to break rules and get away with it.

to skip the computer sessions. Manual scheduling may work for a while, but eventually some signal event will arise for which optimization prescribes a completely unforeseen alternative, deals with a new bottleneck, or otherwise renders some insight. Conversely, schedulers know how to break rules and get away with it. The goal is synergism among the schedulers and their decision-support system, not antagonism.

Appendix

PROFITS generates production campaigns (binary ILP columns) as follows. Each candidate campaign is defined by a production center, a start time, and a sequence of items and production batch quantities.

Index use:

i—item (for example, a product-package combination).

$q = 1,\ldots, Q$—production batch quantity increment.

$w \in W$—work center (for example, producer, conveyer, or packager).

$m(w)$—type of work center w (for example, crew type).

$wp \in \{W \times W \ldots\}$—production center (for example, a combination of work centers).

$t1$, $t2$, $t3$, $t4$, $t5$—times of state reviews: epoch boundaries. For example, $t1$ planning horizon, $t2$ production-demand-inventory state, $t3$ crew constraints, $t4$ work-center utilization, and $t5$ planned shipment.

c—production campaign (binary ILP column).

$s = 1,\ldots, S$—sequence of item batches in a production campaign.

Input data and [units]:

$minibatch_i$—minimum production batch size of item i [quantity].

$batchmultiple_i$—production batch size multiple for item i [quantity].

$maxbatch_i$—maximum production batch size of item i [quantity] (that is, $(maxbatch_i - minbatch_i)$ mod $batchmultiple_i \equiv 0$, and $q = 1,\ldots, Q \equiv 1 + (maxbatch_i - minbatch_i)/bathmultiple_i$).

$time_{wp,i,q}$—time for production center wp to produce item i, quantity increment q [time] (that is, the production rate of $batchmultiple_i/time_{wp,i,q}$ for quantity increment q may express learning effects with batch duration).

$cost_{wp,i,q}$—cost for production center wp to produce item i, quantity increment q [$].

$co\_time_{wp,i,i'}$—changeover time for production center wp from item i to new item i' [time].

$co\_cost_{wp,i,i'}$—changeover cost for production center wp from item i to new item i' [$].

Production campaign definition:

$c : \{wp, t4, \{i, q\}_s, s = 0,\ldots, S\}$—a candidate production campaign uses production center wp, begins at the start of time epoch $t4$, and produces a sequence of batches $\{i, q\}_s$, $s = 1,\ldots, S$, with batch s consisting of item $i[s]$ and batch quantity $q[s]$. Batch 0 represents the production-center state immediately preceding the campaign: If $i[0] = \varnothing$, just set up $i[1]$; an item-to-item changeover is unnecessary.

The duration for a production campaign is planned to be:

$$campaign\_duration = \sum_s \left(co\_time_{wp,i[s-1],i[s]} + \sum_{q' \le q[s]} time_{wp,i[s],q'} \right) + co\_time_{wp,i[S],\varnothing}.$$

At any point in time, the state of a production campaign is characterized by what (if anything) is being

produced, the rate (if any) of production, and the rate of cost expenditure. Production rates and cost rates are assumed to be constant during each production batch quantity increment or during any changeover.

Accordingly, to trace campaign production and cost over continuous time, one need only record the discrete time of each event that changes a state. These event times and state changes are easily enumerated by recording the term-by-term contributions and states of the partial sums accumulated left-to-right in the expression above.

To evaluate a candidate production campaign, sort a list of state-changing campaign events by time over epoch $t1$ along with epoch events over scales $t2$, $t3$, $t4$, and $t5$.

Proceeding over this ordered event list, it is straightforward to accumulate event by event the output data for production campaign c (from which coefficients of binary ILP column c are derived):

$make_{c,i,t2}$—campaign c production quantity of item i during epoch $t2$ [quantity].

$work_{c,m,t3}$—crew for campaign c on work center type m during epoch $t3$ [people].

$dock_{c,i,t5}$—campaign c production quantity of item i during epoch $t5$ [quantity].

$cost_c$—cost of production campaign c [$].

$\{c\}_{w,t4}$—campaigns c that use work center w during epoch $t4$.

When we have too many candidate campaigns, we filter them to select a good subset of reasonable size. Sieves consider the most likely production increments given the demand and inventory state over the planning horizon; make sure that each candidate item appears in a representative number of campaigns; favor campaigns with advantageous setups, changeovers, and shutdowns; concentrate on campaigns that produce items in critically short supply; follow expert advice that good campaigns have these properties; and so forth.

The PROFITS integer linear program selects an optimal portfolio of production campaigns from all admissible candidates:

Index use:

i—item.

$w \in W$—work center (for example, producer, conveyer, or packager).

m—crew type of work center.

$t1$, $t2$, $t3$, $t4$, $t5$—times of state reviews: epoch boundaries. For example, $t1$ planning horizon, $t2$ production-demand-inventory state, $t3$ crew constraints, $t4$ work center utilization, and $t5$ planned shipment.

c—production campaign (binary ILP column).

$\{c\}_{w,t4}$—campaigns c that use work center w during epoch $t4$.

Input data and [units]:

$demand_{i,t2}$—demand for item i during epoch $t2$ [quantity].

$make_{c,i,t2}$—campaign c production of item i during epoch $t2$ [quantity].

$minstock_{i,t2}$—minimum inventory of item i at end of epoch $t2$ [quantity].

$space_i$—storage capacity per quantity i [space/quantity].

$maxspace_{t2}$—maximum inventory at end of epoch $t2$ [space].

$crew_{m,t3}$—crew type m available during epoch $t3$ [people].

$work_{c,m,t3}$—crew for campaign c on work center w during epoch $t3$ [people].

$shipment_{i,t5}$—planned shipment of item i at end of epoch $t5$ [quantity].

$dock_{c,i,t5}$—campaign c production quantity of item i during epoch $t5$ [quantity].

$cost_c$—cost of production campaign c [$].

Decision variables:

$SELECT_c$—1 if campaign c selected, 0 otherwise.

Formulation

subject to

$$\sum_{c,t2' \le t2} make_{c,i,t2'} SELECT_c \doteq \sum_{t2' \le t2} demand_{i,t2'} \quad \forall i,\ t2, \tag{1}$$

$$\sum_{c,t2' \le t2} make_{c,i,t2'} SELECT_c - \sum_{t2' \le t2} demand_{i,t2'} \doteqdot \text{minstock}_{i,t2} \quad \forall i,\ t2, \tag{2}$$

$$\sum_{i,c,t2' \le t2} space_i make_{c,i,t2'}\ SELECT_c - \sum_{i,t2' \le t2} space_i demand_{i,t2'} \doteqdot \text{maxspace}_{t2} \quad \forall t2, \tag{3}$$

$$\sum_{c} work_{c,m,t3} SELECT_c \doteqdot crew_{m,t3} \quad \forall m,\ t3 \tag{4}$$

$$\sum_{\{c\}_{w,t4}} SELECT_c \le 1 \quad \forall w,\ t4, \tag{5}$$

$$\sum_{c,t5' \le t5} dock_{c,i,t5'} SELECT_c \dot{\geq} \sum_{t5' \le t5} shipment_{i,t5'} \quad \forall i,\ t5, \tag{6}$$

$$SELECT_c \in \{0, 1\} \quad \forall c, \tag{7}$$

$$\underset{SELECT}{\text{minimize}} \sum_c cost_c SELECT_c + \text{elastic penalties}. \tag{8}$$

For each time epoch $t2$, each constraint (1) expresses that cumulative production of item i should equal cumulative demand by the end of that epoch and (2) additionally stipulates that cumulative production of item i less cumulative demand by the end of that epoch should meet a minimum inventory goal. The symbols $\dot{\leq}$, $\doteq$, and $\dot{\geq}$ denote elastic constraints: For instance, constraint (1) charges a linear penalty per unit violation below its cumulative demand goal (unmet demand) or a different linear penalty per unit violation above this goal (carrying inventory). Brown et al. (1997) further motivate cumulative goals and this presentation style.

Each elastic constraint (3) expresses the maximum storage goal at the end of epoch $t2$. During each epoch $t3$, an elastic constraint (4) limits work-center-type m crew utilization. During each epoch $t4$, an inelastic packing constraint (5) limits the utilization of each work center w. By the end of epoch $t5$, an elastic constraint (6) tries to accumulate sufficient production of item i to meet a scheduled shipment. Campaign selections are binary (7).

The objective function (8) expresses for planning horizon epoch $t1$ the total cost of all selected production campaigns plus any penalties for violating elastic constraints.

This model is solved with an algorithm (that is, Brown and Olson's [1994]) that expresses the elastic penalties logically so that the model appears to the solver to be expressed exclusively in terms of binary variables.

This model can be extended to accommodate multistage production. The best alternative for multistage production depends upon whether intermediate items can be stored in inventory or not, whether the stages are serial or only partially ordered, and whether production uses fixed recipes. For instance, in our experience, pharmaceutical production involves a sequence of brewing and purifying batch operations over a fairly long time horizon. This type of production can be well expressed by having each item induce a demand for its predecessor items and by letting demand for the finished item attract the optimal preceding production-scheduling chain. Within each type-two epoch, at most one stage of batch production is active. Petrochemical refining and blending are similar but may employ continuous-process production campaigns that run for extended periods, consuming multiple intermediate input items and producing multiple output items. Type-two events update inventory resulting from production and external demand and from internal process consumption and production of all affected items. A job shop has many items, each requiring a particular sequence of operations on the production facilities. Here, simultaneous production campaigns each consist of exactly one setup, production run and shutdown, type-four epochs deconflict simultaneous demands for individual production lines, and each production line is used for at most one item in any type-four epoch.

References

Brown, G. G., R. F. Dell, R. K. Wood. 1997. Optimization and persistence. *Interfaces* **27**(5) 15–37.

——, A. M. Geoffrion, G. H. Bradley. 1981. Seasonal production and sales planning with limited shared tooling at the key operation. *Management Sci.* **27**(3) 247–259.

——, M. P. Olson. 1994. Dynamic factorization in large-scale optimization. *Math. Programming* **64**(1) 17–51.

Clark, W. 1922. *The Gantt Chart*. Ronald Press, New York.

Geoffrion, A. M., G. W. Graves. 1976. Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach. *Oper. Res.* **24**(4) 595–610.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Boston, MA.

Hoffman, K. L., M. Padberg. 1993. Solving airline crew scheduling problems by branch-and-cut. *Management Sci.* **39**(6) 657–682.

Kang, S., K. Malik, L. J. Thomas. 1999. Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. *Management Sci.* **45**(2) 273–289.

Kuik, R., M. Salomon. 1990. Multi-level lot-sizing problem: Evaluation of a simulated-annealing heuristic. *Euro. J. Oper. Res.* **45**(1) 25–37.

Lawrence, K. D., S. H. Zanakis. 1984. *Production Planning and Scheduling: Mathematical Programming Applications*. Industrial Engineering and Management Press, Norcross, GA.

Nicholson, T. A. J., R. D. Pullen. 1971. A linear programming model for integrating annual planning of production and marketing. *Internat. J. Production Res.* **9**(3) 361–369.

PowerVista. 2001. PowerVista Bridge Graphical Application Toolkit. Conifer, CO ⟨www.powervista.com⟩.

Sibre, C. E. 1977. A quadratic assignment/linear programming approach to ship scheduling for the U.S. Coast Guard. M.S. thesis, Naval Postgraduate School, Monterey, CA.